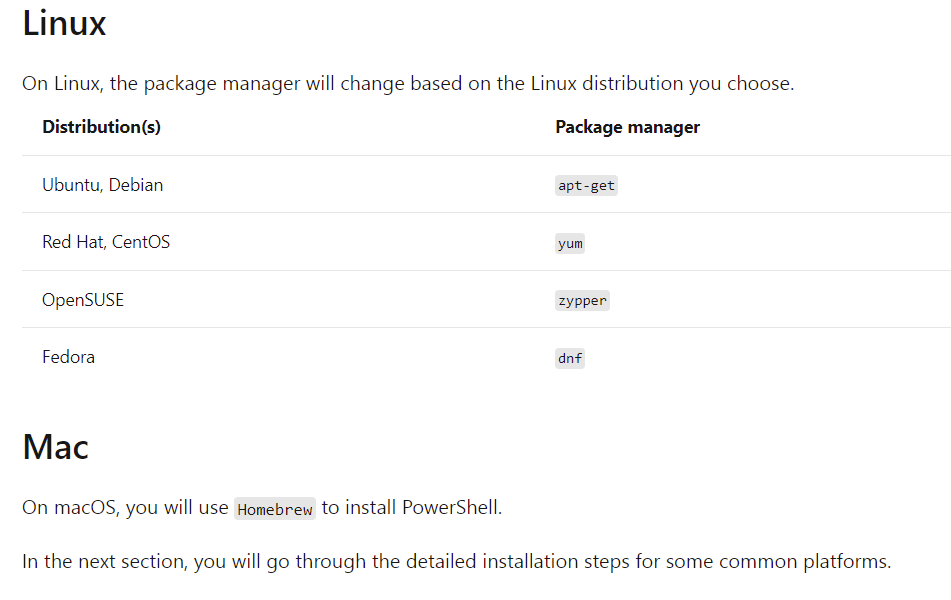
1. Installing Azure PowerShell:

#Checking PowerShell version essential for Azure

[PowerShell]$PSVersionTable.PSVersion

If Major < 5

#Need to get PowerShell 7.0.6LTS, PowerShell 7.13

https://docs.microsoft.com/en-us/powershell/scripting/install/installing-powershell-on-windows?view=powershell-7.2

#Once installed, Powershell.exe

* Pwsh –ver
* Pwsh.exe
* To check powershell version
* Add Dotnet nugget to allow dotnet tool
* dotnet nuget add source --name nuget.org <https://api.nuget.org/v3/index.json>
* dotnet tool install --global PowerShell

Install the Az PowerShell Module:

Install-Module -Name Az -Scope CurrentUser -Repository PSGallery –Force -Verbose

Allow scripts to run:

Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser

Update a module:

Update-Module -Name Az

# Install .NET Core SDK

https://dotnet.microsoft.com/en-us/download

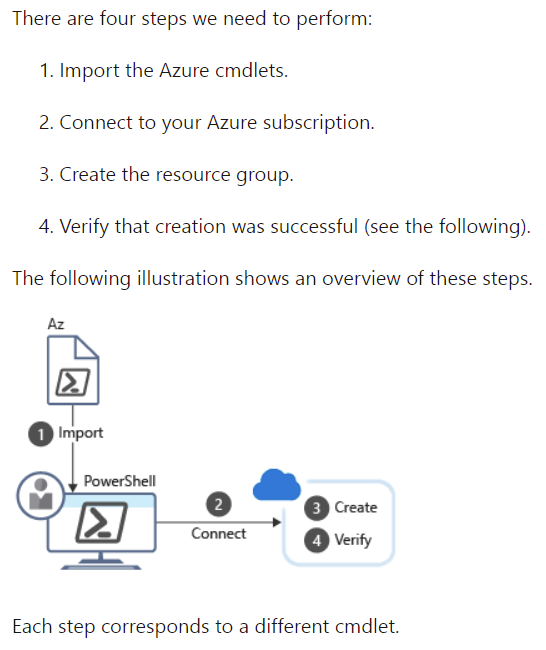
#Once installed

* Cmd
* Dotnet
* Dotnet –info
* To check Dotnet framework version

# Install .NET Global tool

[PowerShell]dotnet tool install --global PowerShell

1. Creating a Resource Group with Azure PowerShell:



1. Connect to Azure Account:

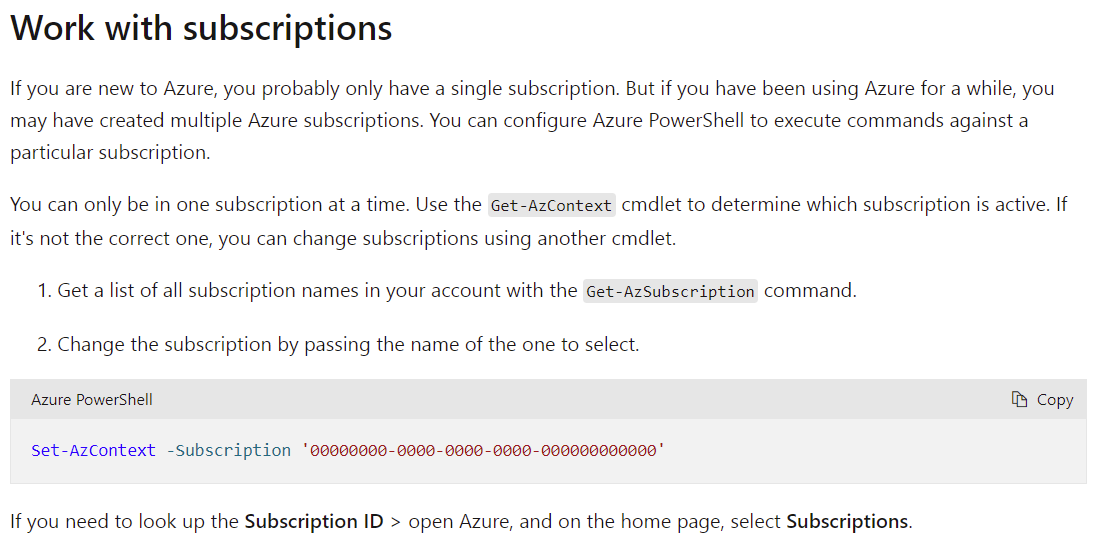
Connect-AzAccount

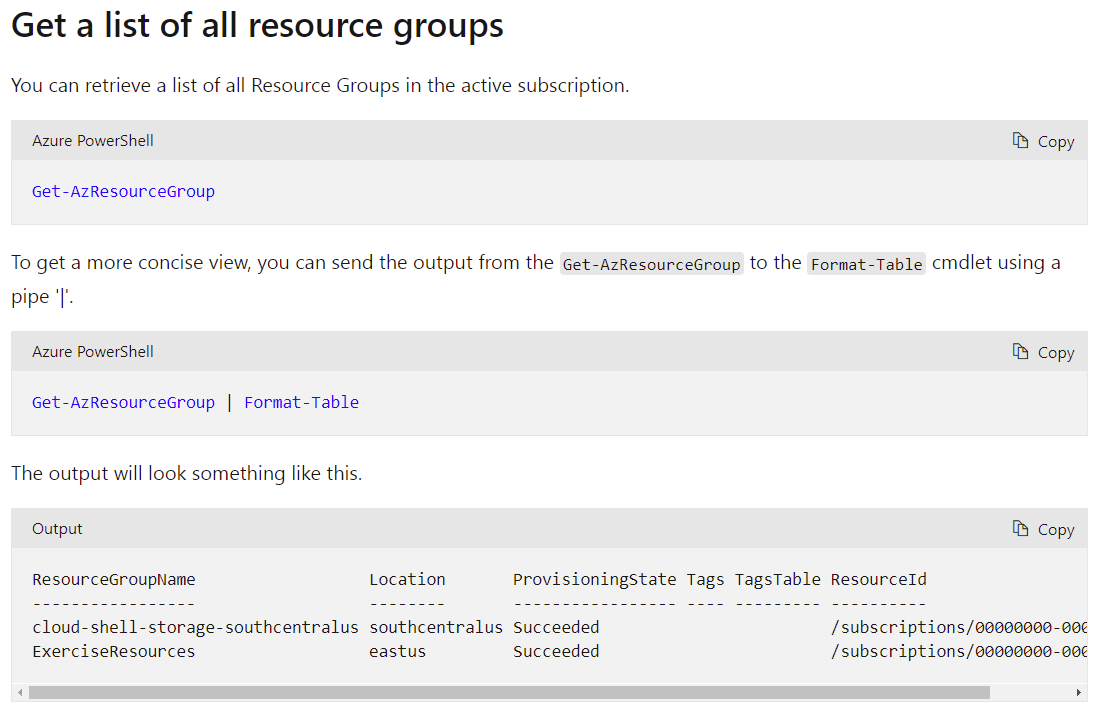
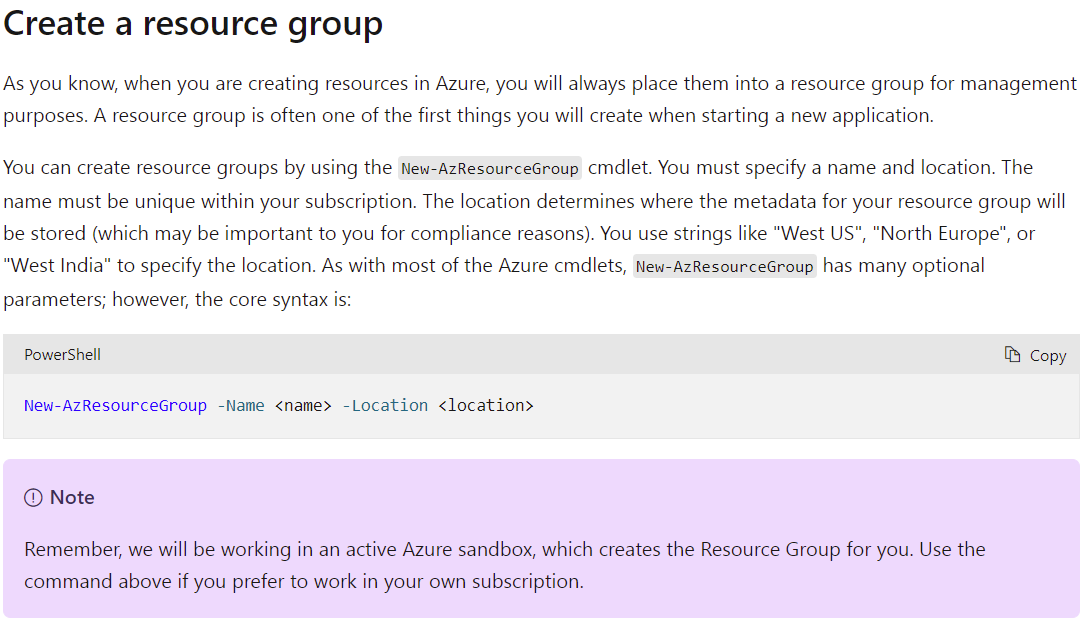
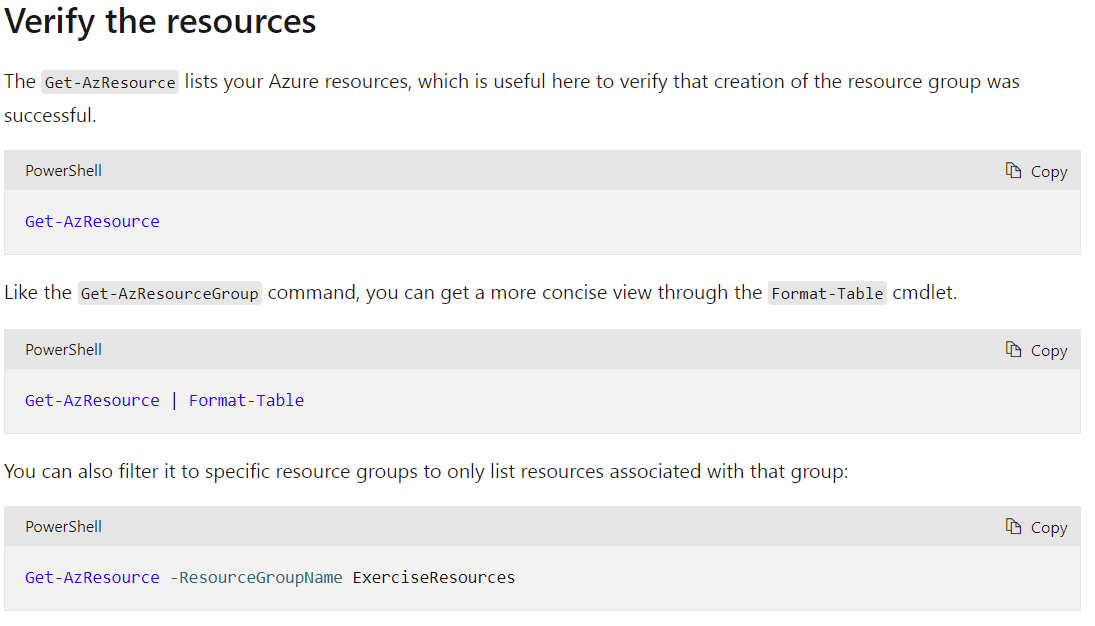
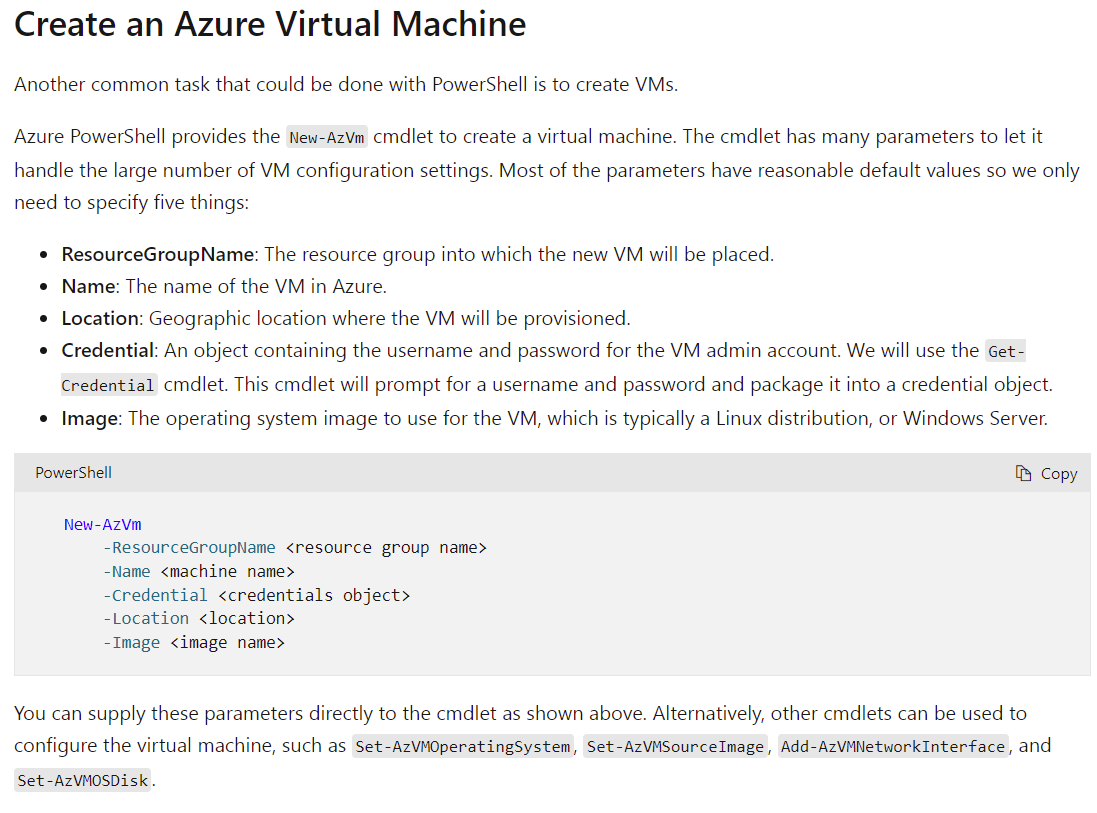
Azure account works with subscription

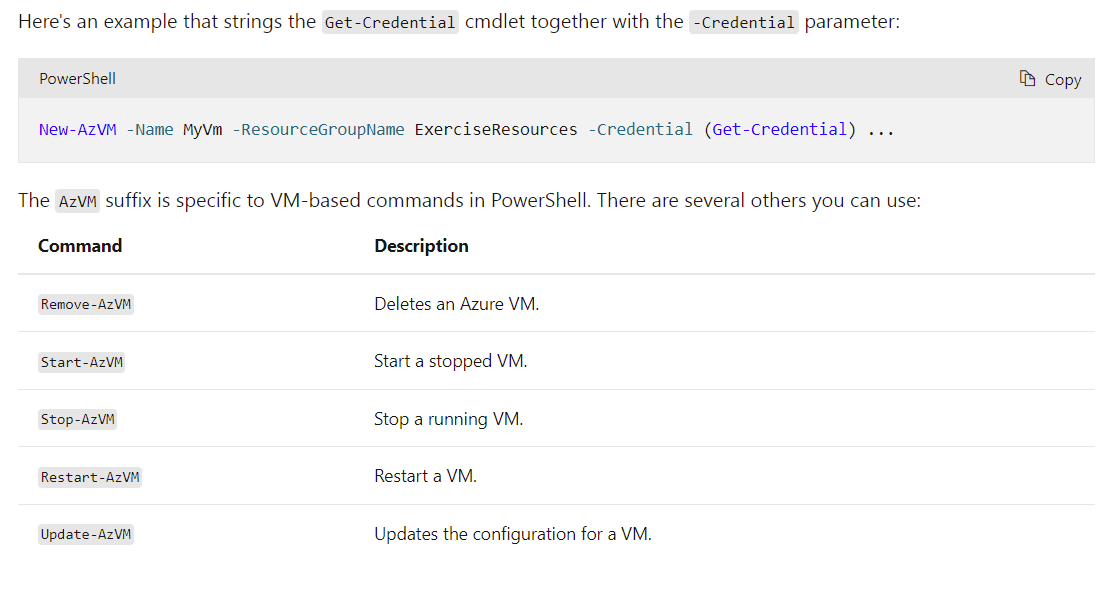
Get-AzSubscription

To check active subscription

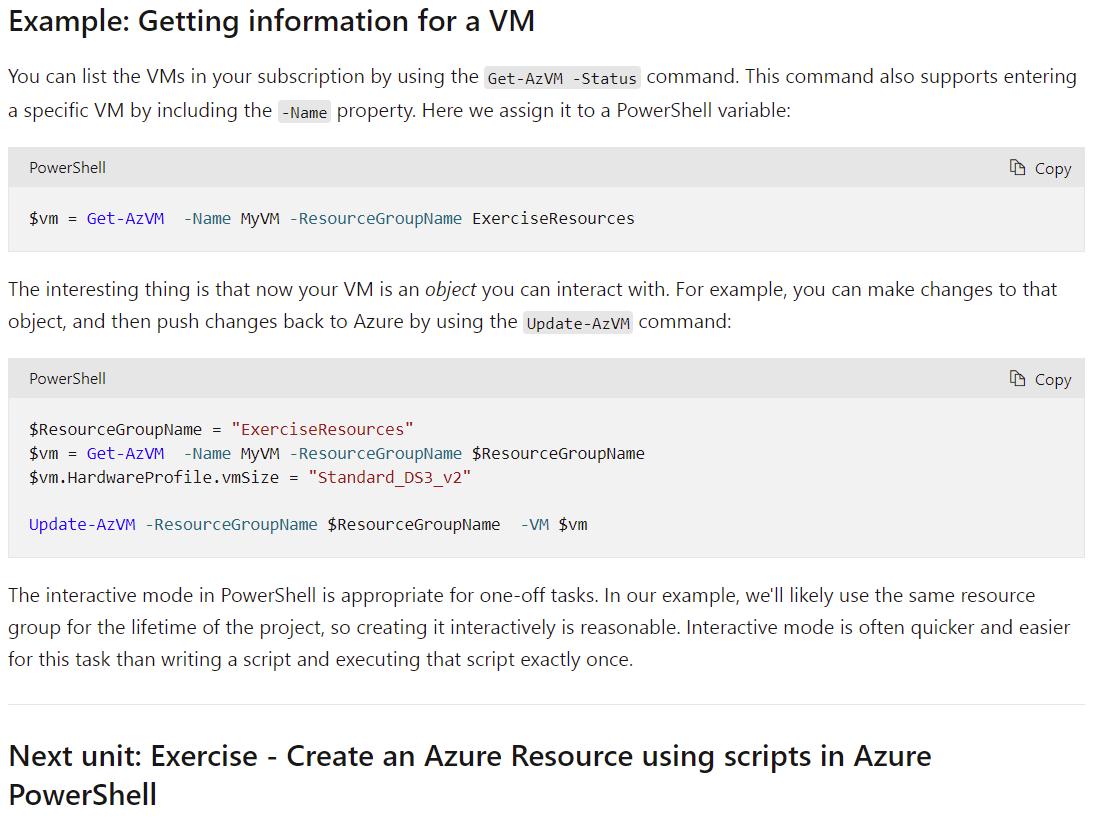
Get-AzContext



1. Get a list of all resource groups:
2. Create a resource group:
3. Verify the resources:
4. Create an Azure VM:
5. Create Azure VM example:



1. Example – Getting information for a VM:

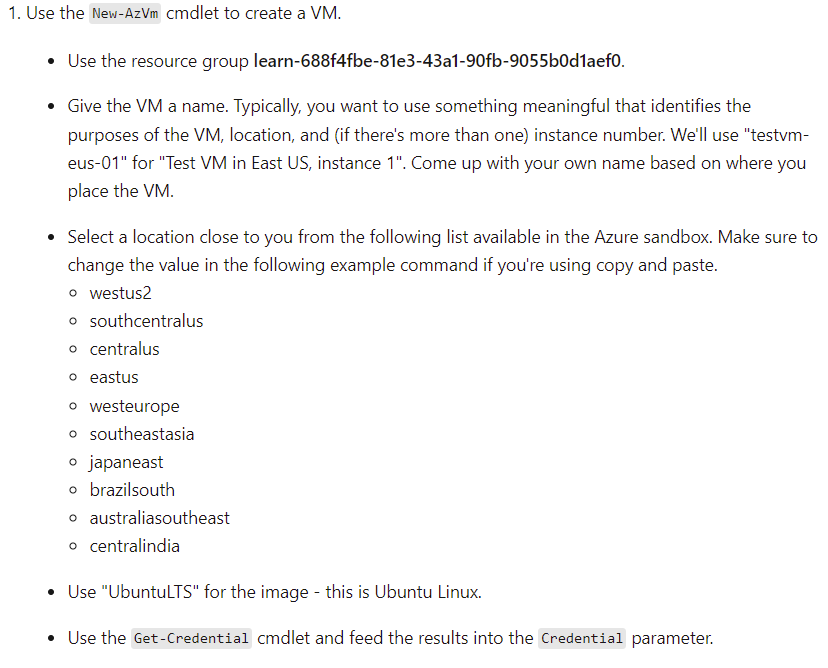


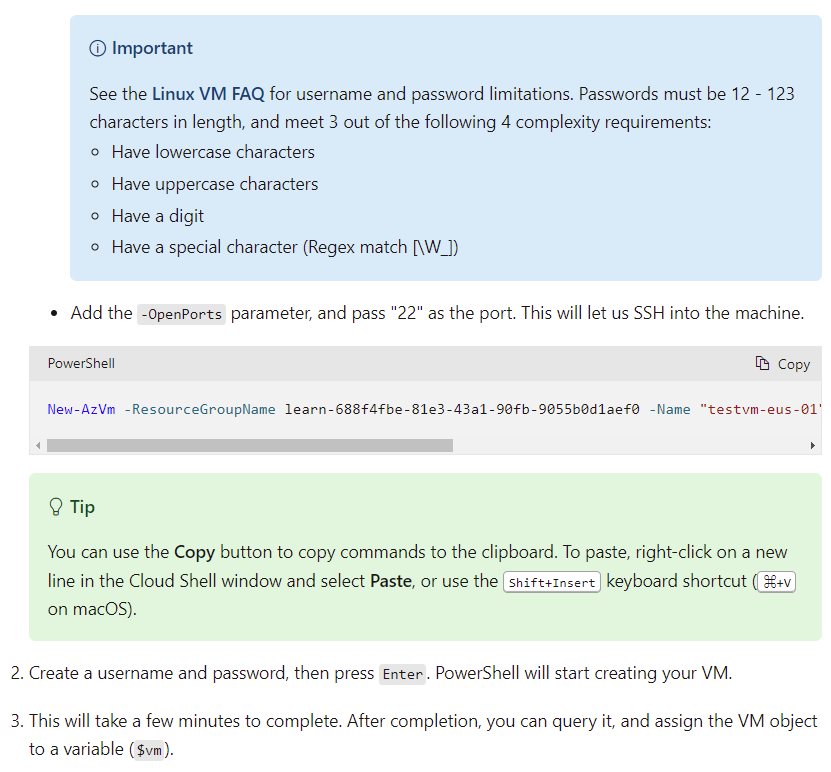
1. Create an Azure Resource using Azure PowerShell:

Sign in Microsoft to enable Sandbox:

<https://docs.microsoft.com/en-us/learn/modules/automate-azure-tasks-with-powershell/6-exercise-create-resource-interactively?activate-azure-sandbox=true>

1. Create an Azure Sandbox Ubuntu Linux VM:



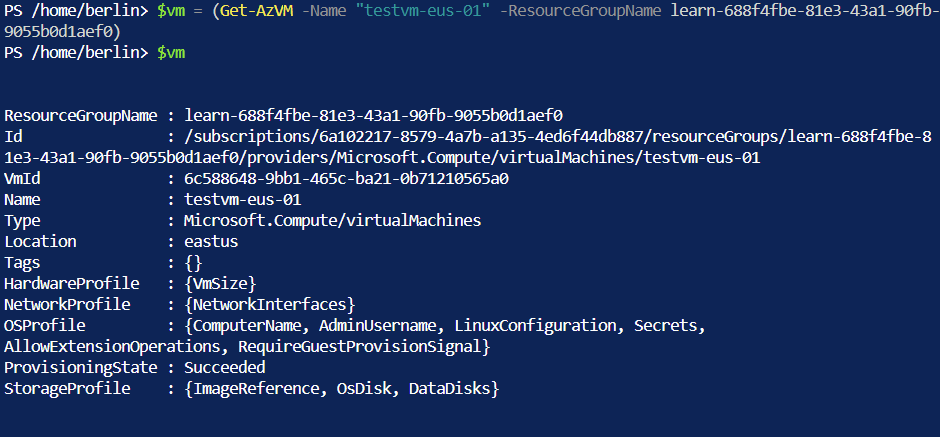


New-AzVm -ResourceGroupName learn-688f4fbe-81e3-43a1-90fb-9055b0d1aef0 -Name "testvm-eus-01" -Credential (Get-Credential) -Location "East US" -Image UbuntuLTS -OpenPorts 22

$vm = (Get-AzVM -Name "testvm-eus-01" -ResourceGroupName learn-688f4fbe-81e3-43a1-90fb-9055b0d1aef0)

$vm

1. Check VM created:



13. Check VM info:

#$vm

$vm = (Get-AzVM -Name "hostname" -ResourceGroupName learn-688f4fbe-81e3-43a1-90fb-9055b0d1aef0)

$vm | gm

#Check VM hardware

$vm.HardwareProfile

#Get information on disks

$vm.StorageProfile.OsDisk

#Check VM Public IP Address

$vm | Get-AzPublicIpAddress

#Connect to an Ubuntu Linux Server

Ssh <username>@<IpAddress>

14. Deleting a VM:

#Shutdown a VM

Stop-AzVM –Name $vm.Name –ResourceGroupName $vm.ResourceGroupName

#Deleting a VM

Remove-AzVM –Name $vm.Name –ResourceName $vm.ResourceGroupName

#List all the resources in your resource group

Get-AzResource –ResourceGroupName $vm.ResourceGroupName | Format-Table

#Deleting VM network interface

$vm | Remove-AzNetworkInterface –Force

15. Hands-on creating a VM using PowerShell scripts:

-> Go to Azure Cloud Shell:

Cd $HOME\clouddrive

# Create a new .txt, ConferenceDailyReset.ps1

Touch “./ConferenceDailyReset.ps1”

# Open the integrated editor, & select ConferenceDailyReset.ps1

Code “./ConferenceDailyReset.ps1”

# A word editor window will pop up

# Start by capturing the input parameter in a variable

Param(

[string]$resourceGroup

)

$adminCredential = Get-Credential –Message “Enter a username & pw for VM administration.”

#Create a loop that executes 3 times.

For (

$i = 1; $i –le 3; $i++

)

{

$vmName = “ConferenceDemo” + $i

Write-Host “Creating VM: “ $vmName

New-AzVm –ResourceGroupName $resourceGroup –Name $vmName -Credential $adminCredential –Image UbuntuLTS

}

# Use “…” on top right corner of editor to save OR

# Ctrl + S

# Close editor

# Run script

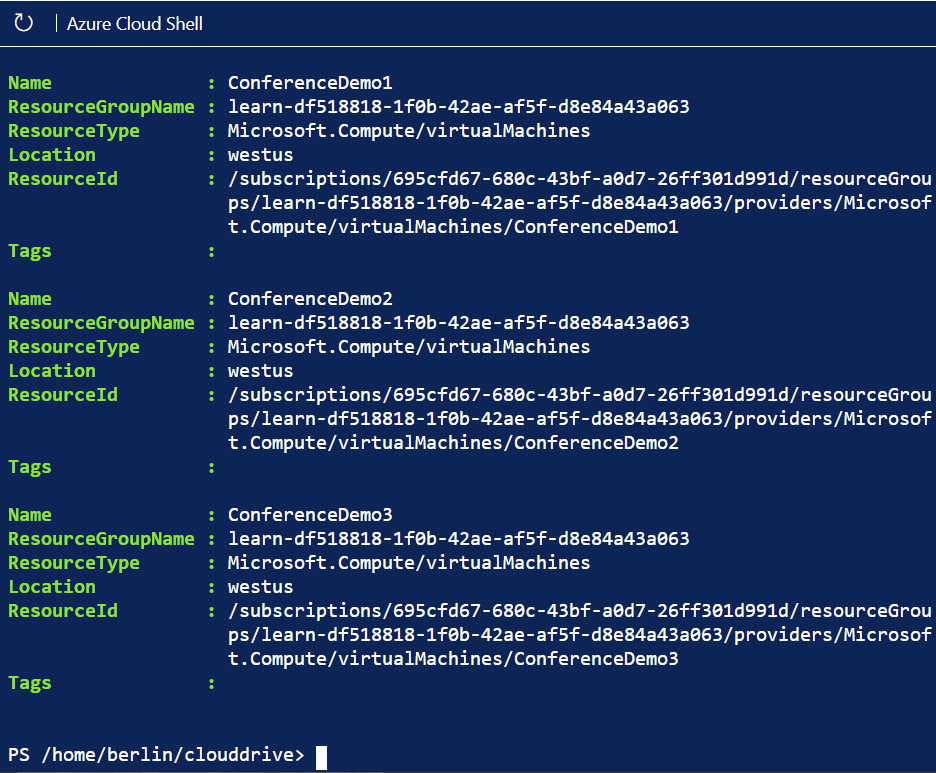
./ConferenceDailyReset.ps1 learn-df518818-1f0b-42ae-af5f-d8e84a43a063

#Enter a username & a strong password

# Check VMs’ existence

Get-AzResource –ResourceType Microsoft.Compute/virtualMachines

# Results:



# Clean up of AzResourceGroup

Remove-AzResourceGroup –Name MyResourceGroupName